



**Calhoun: The NPS Institutional Archive**

---

Faculty and Researcher Publications

Faculty and Researcher Publications

---

2015

# Mobile Cloud Computing for C2 - Operating in DIL Network Conditions

Ling, Yu Xian

---

<http://hdl.handle.net/10945/44718>



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

**Dudley Knox Library / Naval Postgraduate School  
411 Dyer Road / 1 University Circle  
Monterey, California USA 93943**

<http://www.nps.edu/library>

## **20th ICCRTS**

### **Mobile Cloud Computing for C2 - Operating in DIL Network Conditions**

**Primary Topic:** (5) Modeling and Simulation

Alternate Topic 1: (4) Experimentation, Metrics and Analysis

Alternate Topic 2: (6) Cyberspace, Communications and Information Networks

**Paper Number: 032**

**Name of Author(s)**

Yu Xian Ling (*Singapore Defence Science and Technology Agency*)

Toon Joo Wee (*Singapore DSO National Laboratories*)

Man-Tak Shing, Gurminder Singh, John H. Gibson (*Naval Postgraduate School*)

**Point of Contact**

Man-Tak Shing  
Computer Science Department  
Naval Postgraduate School  
1141 Cunningham Road, GE334  
Monterey, CA 93943  
+1 831-656-2634  
shing@nps.edu

## *Abstract*

Cloud computing is emerging as the mainstream platform for a range of on-demand applications, services, and infrastructure. It is already playing an important role in the communication, and the processing, mining, and fusing of information in distributed command and control. A major benefit of cloud computing is improved net-centric capability. Before the full benefits of cloud computing are realized, several technology challenges must be addressed. Operating in intermittent and austere network conditions is one such challenge, which navy ships face when communicating with land-based cloud computing environments.

We investigate the data requirements of navy ships and propose two mechanisms – data caching and cloudlets – to improve the cloud connectivity under intermittent and austere network conditions. We study the application of these two mitigating strategies in detail and evaluate their performance through modeling and simulation for both individual ships as well as ships in a Carrier Strike Group or an Expeditionary Strike Group (CSG/ESG). Results from our simulations have suggested a positive impact. Caches and cloudlets as a part of the shipboard architecture produce better performance in data communications. Most importantly, the strategies promote operations continuity for a naval force under disconnected, intermittent, and limited (DIL) network environments.

**Keywords** — *mobility, cloud computing, cloudlet, cache, DIL, command and control*

## **I. Introduction**

The U.S. Navy (USN) and its coalition partners have become increasingly dependent on the availability of stable and robust ship-to-shore satellite communication (SATCOM) to deliver network and application services [8]. While SATCOM has provided unprecedented support for military services, the communication supported by SATCOM is less than reliable in terms of its quality of service and connectivity. Command and control (C2) is one of the services that will be greatly affected by intermittent communication. Tactical situations increase the likelihood of a disconnected, intermittent, and low-bandwidth (DIL) environment while simultaneously increasing the need for an updated and synchronized common operational picture (COP) [10]. Existing C2 systems using event-based protocols to manage tracks may conserve bandwidth, but they do not guarantee a common operating picture in DIL environments. Without an updated COP among the involved parties, confusion can arise among them, and well-informed and sound decisions cannot be made. Given limited bandwidth and intermittent connectivity of satellite connections, new architectures are needed to support data requirements of navy ships. Motivations for the research efforts have derived not only from concerns about the risk that satellites can be jammed or even shot down during hostilities, but also concerns about the cost and availability of satellites world-wide and to all partners in potential coalitions even in peacetime.

Cloud computing is emerging as the mainstream platform for a range of on-demand applications, services, and infrastructure. It is already playing an important role in the communication, and the processing, mining, and fusing of information in distributed command and control. A major benefit of cloud computing is improved net-centric capability. In most cloud-based systems, clients generate and/or consume information, and are connected to cloud-based servers over wired or wireless network connections. For mobile clients, this connection, by necessity, is a wireless connection. While cloud computing has brought about unprecedented sophistication in the mobile ecosystem, there are a number of challenges that need to be addressed in order for the overall environment to be dependable. Operating in intermittent and austere network conditions (fluctuating wireless bandwidth, intermittent connectivity, and reliable connectedness of mobile clients) is one such challenge, which navy ships face when communicating with land-based cloud computing environments.

This paper provides an extended summary of our research on strategies to improve interactions between mobile platforms and the cloud under intermittent and austere network conditions [16]. It reviews the current navy shipboard data usage and examines two mitigating strategies – data caching and cloudlets [11]. The rest of the paper is organized as follows. Section II gives an overview of the current shipboard data usage. Sections III and IV discuss the cloud response time and present strategies to overcome high latency and intermittent connections. Section V presents an analysis of our proposed strategies for individual ships as well as ships in a Carrier Strike Group or an Expeditionary Strike Group (CSG/ESG). Section VI discusses the findings of the study and Section VII draws some conclusions.

## **II. Shipboard Data Usage**

There are four main categories of shipboard data usages: Command and Control (C2) data, Positioning Navigation and Timing (PNT) data, Meteorological and Oceanographic (MTEOC) data, and Quality-of-Life (QoL) data.

#### **A. C2 Data**

Situational awareness is a vital function of a C2 system. The quality of a commander's decision for the next course of action greatly depends on the accuracy and timeliness of the C2 data provided by the system. C2 data are made up of the following data types: text/tracks (e.g. C2 messages, tasking orders, status updates, ISR reports), images (e.g. ISR images, sensor feeds, map overlays), and videos (e.g., ISR videos, UAV feeds). To accomplish a sufficiently effective situational awareness, the C2 system must fuse and display these data, which can be obtained from sensors, human intelligence, signal intelligence, communications intelligence, image intelligence, or even open-source intelligence, in a clear and intuitive manner.

#### **B. PNT Data**

Positioning Navigation and Timing (PNT) distribution systems are required to provide a common geospatial platform and temporal reference to military platforms. This data is pervasive and critical for military platforms, because it supports many targeting, situational awareness, communication, and weapon systems. Overall mission effectiveness is also highly dependent on PNT data [9]. The Navigation Sensor System Interface (NAVSSI), being the primary source of PNT data, gathers inputs from multiple shipboard sensors and then distributes the resultant navigation, time, and frequency data to both internal and external systems for consumption. Time criticality is the other important factor in distributing PNT data to naval systems.

#### **C. METOC Data**

Weather conditions in both the atmosphere and ocean can affect how the U.S. Navy carries out their operation. It is difficult to make an accurate prediction of the weather, and this impedes the naval forces from planning and executing their mission efficiently and effectively. Force structure composition, force movement prediction, personnel safety, estimation of capability performance, and war-fighting tactics are examples of what the adverse weather can impact. Hence, the need for meteorological and oceanographic (METOC) information is critical.

#### **D. QoL Data**

Conducting official and personal business via the Internet is a basic necessity in modern-day life. Besides conducting Navy business, sailors afloat need Internet support for their continuing education, banking, daily news update, entertainment, social networking, and family contacts. Although Internet support for personal QoL data may not be mission-critical, its quality of service has direct impact to the overall morale of the personnel afloat.

### **III. Cloud Response Time**

For real-time and highly interactive applications, fast response time is a key requirement for satisfactory performance. In [6], Cloud Mobile Gaming (CMG) and Cloud Mobile Desktop (CMD) applications are used to investigate the viability of using a public cloud server provider, Amazon Web Services (AWS). Table 1 shows the response time requirements based on a user survey. Figure 1 shows the average response times for both 3G and WiFi for the streaming of video using the commercial video conferencing software, Skype. Figure 2 shows the average response time for viewing slide shows and typing using the remote desktop application, Citrix. We can see that the average response time for both 3G and WiFi are higher than the acceptable range in all cases, which means that it would not be a satisfactory user experience.

Table 1. Response Time Requirement for CMG and CMD (after [6]).

	CMG	CMD	
		Slide Show	Typing
Acceptable	440 ms	835 ms	390 ms
Excellent	280 ms	445 ms	125 ms

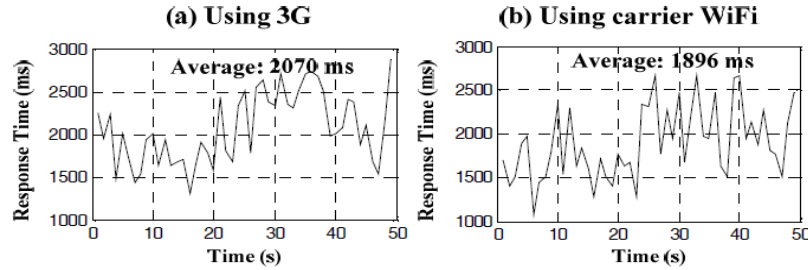


Figure 1. Response time using Skype to stream CMG video (from [6]).

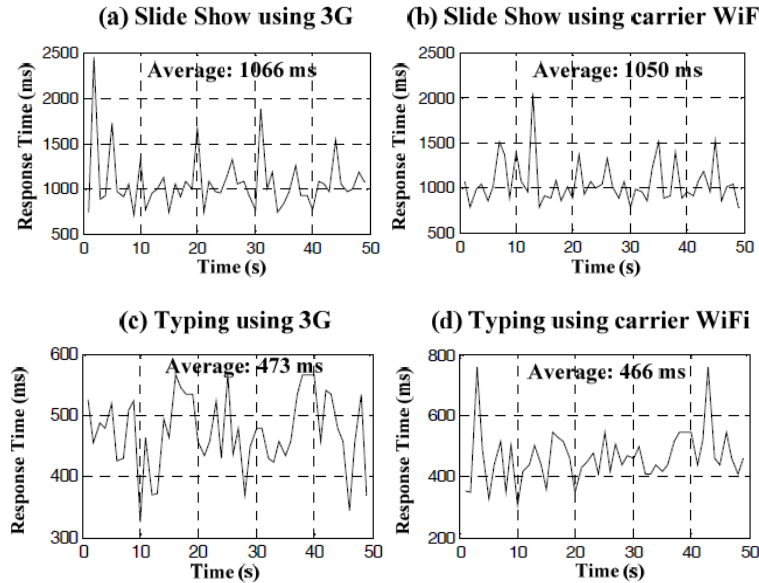


Figure 2. Response time using Citrix as CMD application (from [6]).

In [1], a series of experiments was conducted on Amazon Elastic Compute Cloud (EC2) to study the response time of five types of Amazon EC2 instances, with different types of virtual machines in terms of CPU capacity, RAM size, and disk size. As expected, the result showed that we can get faster and more stable response time with better CPU capacity, more RAM, and larger disks.

In [5], Chen investigated the upload/download speed between different AWS regions to see the differences in speed when transmitting data between EC2 and Simple Storage Service (S3) buckets in different regions. The result shows that the best upload time occurred when both the EC2 instance and S3 bucket were located in the same region.

The above results help reinforce our belief that ample local storage and physical location to the data sources is very important to improve the communications. Amazon provides the option, called Amazon CloudFront [2], for businesses and developers who want to distribute

content to end users with low latency and high data transfer speeds. Amazon CloudFront is a content distribution network (CDN) which is tightly integrated with Amazon S3. It is designed specifically to improve static content delivery. S3 is designed to easily store and retrieve data. When S3 is used together with CloudFront, S3 becomes the offsite backup of CloudFront. CloudFront moves the S3 content to the network “edge,” geographically closer to the end user, which helps reduce latency as shown in Figure 3. It is a pull model where content is pulled from S3 to the edge upon first request and it expires in 24 hours by default.

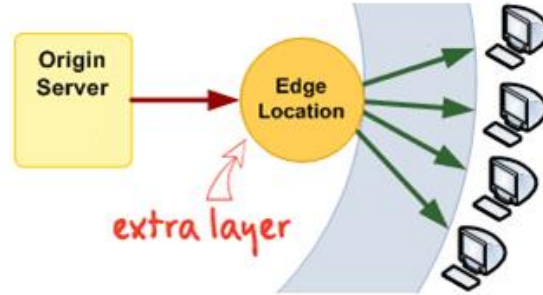


Figure 3. Illustration of an edge cache nearer to end users (adapted from [3]).

The Amazon CloudFront is similar to the cloudlet concept, which was introduced by Satyanarayanan and his colleagues in [11] to overcome the high latency in cloud networks. Being a resource-rich element that has good connectivity to the Internet and mobile devices, cloudlets allow low end-to-end latency to be achieved. This is analogous to Wi-Fi access points which are in close proximity to a user’s mobile device, allowing the mobile device to enjoy a higher signal strength and higher speed access to the Internet. The physical proximity of cloudlets proves advantageous to serving hostile environments that are attributed with short-term large-magnitude uncertainty [12]. Instead of relying on a cloud that is far away and being susceptible to poor connectivity, cloudlets can provide a closer and resource-rich alternative. Other subtle benefits of having cloudlets include safe deployment in insecure areas such that tampering, loss, or destruction of cloudlets do not prove to be a major security issue. This is due to the content of cloudlets being in soft states only.

#### IV. Strategies To Overcome High Latency and Intermittent Connections

We want to leverage the benefits of caching to support real-time and near real-time data usage. With caches, data that has been previously requested can be stored locally and the next time this data is requested again, it will be more readily available. Therefore, one of our proposed strategies is the implementation of caches.

Caches are deployed on each node to facilitate the requests made by each node to the remote cloud server. When a node requests certain data, it will first look at its own cache. If the data is not available on the local cache, it will make a request for the required data from the cloud server. Once this data is retrieved, it will be stored in the local cache of the requesting node. The next time the same data is requested, it will be available in the local cache and this shortens the overall response time, thus improving the performance of the data transaction.

In many cases, it is not feasible to create a cloud infrastructure that is within the range of every node. Therefore, we will want to deploy cloudlets to be within close proximity of every

node, to close the gap by extending each node's maximum connectivity range. This way we can leverage the cloudlets' connectivity to the remote cloud server. Each cloudlet is assumed to be deployed on the large platform (referred to as the cloudlet node) of a CSG/ESG, although any node can also take on the role of a cloudlet node. A cloudlet node is responsible for the communication back to the remote cloud server ashore for data access. The other naval platforms (commonly referred to as nodes) can connect to the cloudlet node to access information that they require. In addition, cloudlets are incorporated with caches to supplement the nodes connecting to cloudlets to access information, and thereby support real-time and near-real-time cases. If data is not available in the cache of the cloudlet node, the cloudlet node can request the information from the remote cloud server on behalf of the nodes via the satellite.

## V. Simulation Study of the Mitigation Strategies

The objective of our analysis is to test whether the implementation of caches will provide benefit in a DIL environment. Intuitively, the volume of data, the bandwidth of the communication link, and the response time of our data traffic (from source to destination and back) are directly related to one another. For example, with a fixed amount of bandwidth, the higher the data volume, the longer it will take for the source node to receive a reply from its destination. Similarly, the lower the data volume, the faster will be the response time. That is, as the offered load, or traffic intensity, increases so too does the response delay, generally due to increases in queuing delays throughout the system.

Cache performance [7] is generally measured using average memory access time (AMAT) as follows:

$$AMAT = hit\ time + miss\ rate \times miss\ penalty \quad (1)$$

In order to fit our requirement, we need to relate this formula with the consideration of the volume of data, the bandwidth of the communication link, and the response time of our data. The following paragraphs step through the process of deriving a formula that measures the response time for our model.

In web caching, there are generally three kinds of data, static, semi-static, and dynamic. They are categorized based on lifetime of the data, or Time-To-Live (TTL).

- Static: the data does not change in its lifetime (TTL = infinity). For example, a static web page with no dynamic content. The data does not change for every request, thus, caching is most useful for this kind of data.
- Semi-static: the data does change but not that often ( $0 < TTL < \text{infinity}$ ). For example, weather forecast webpages that are updated every two hours. The data does change for some requests; thus, caching is still useful but not as much as for static data.
- Dynamic: the data changes for every request (TTL = 0). For example, a real-time stock price webpage that presents different information every second or less. Caching dynamic data is the least useful.

We want to model our operating environment as close to the current naval environment as possible, but due to the security classification of the existing data set, we were only able to take reference from public sources. Our data usage profile will be based on actual Internet traffic,



with reference from Sandvine [13], a broadband equipment company. This will give us an approximate representation of the existing data usage profile of the U.S. Navy QoL traffic. However, by plugging in accurate navy requirements for other data types, we can get navy-specific results for C2. Sandvine's bi-annual report measures the average Internet traffic demand of a general Internet user for the first half of 2014, and it also provides a categorical breakdown of the traffic demand as shown in Figure 4.

Monthly Consumption - North America, Fixed Access		
	Median	Mean
Upstream	1.4 GB	7.6 GB
Downstream	17.4 GB	43.8 GB
Aggregate	19.4 GB	51.4 GB




Figure 4. Monthly Consumption Figures (per individual user) – North America, Fixed Access (from [13]).

We reckon that upstream consumption is probably the data demand for uploading. An example of such would be situation reports or intelligence collection data. Another potential source would be video-teleconference streams. However, it is expected the bulk of the upstream traffic will be requests for data and as such it is also expected that the downstream traffic will dominate the sessions. Since upstream consumption is small as compared to downstream, it is reasonable to use aggregated data consumption as our total data volume. From Figure 4, we have the average monthly consumption of data at 51.4GB. Hence, the total data demand rate for our test set is calculated to be approximately 21,300 bytes per second per user (this number is a simple conversion of the data demand from month to seconds).

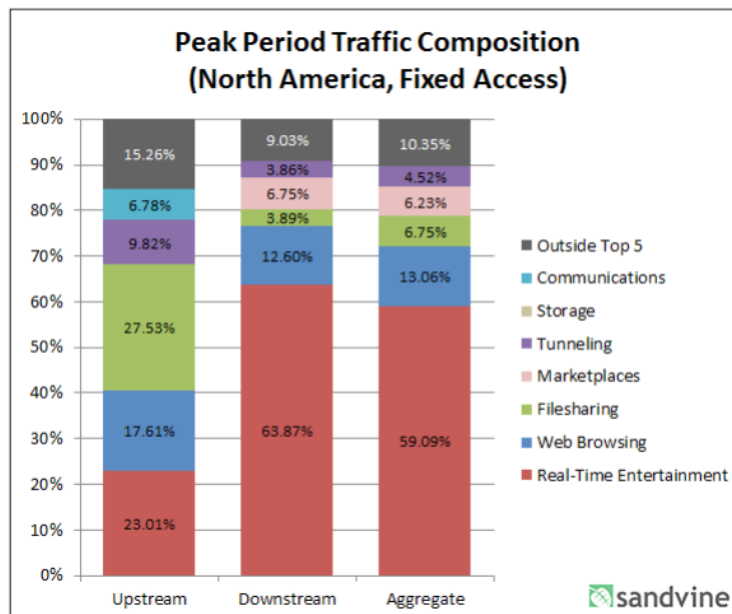


Figure 5. Peak Period Aggregate Traffic Consumption – North America, Fixed Access (from [13]).

In Figure 5, we see that real-time traffic, covering applications which require “on-demand” data, takes up about 59 percent of the total demand. Communications traffic, consisting of real-time chat, voice, and video communications, takes up 13 percent of the total traffic demand. These two categories of data fall under the data type of dynamic data, as their content are continuously updated, making them non-cacheable. The rest of the categories will be broadly categorized into cacheable type of data, taking up 28 percent of the total data demand. In another study, Wessels [17] reports that between 35 and 70 percent of all requested objects are cacheable for general Internet traffic. To understand how the ratio of cacheable objects affects the response time in our experiment, we intend to run our model over the range of 30 to 70 percent of cacheable data. This will give us a good coverage of data with the characteristics of being cacheable and non-cacheable.

Web caching [4] can provide significant benefits to both the end user and the service provider. The end user can enjoy a faster surfing of the web if the requested objects are in the cache. For the service provider, there will be savings in the bandwidth. The mentioned benefits can be achieved only when the requested objects from the web are available in the cache. This is the probability of the requested objects being found in the cache, which is called probability of a hit or hit ratio,  $P(\text{hit})$ . The hit ratio is dependent on several factors, such cache size, number of objects available in the Internet, average size of object, and percentage of cacheable objects, etc. Another key factor is the degree to which the requested data relates to other data, what might be referred to as data cohesion. One of the tenets of cache regards special locality of reference, that is, a reference to data at one location is likely to coincide with a reference to data that is located nearby. When considering the data use requirements of vessels within a strike group it is reasonable to expect that if one ship requires a given data set others are likely to also require it, thus subjectively substantiating the utility of a cloudlet on the flagship to service the rest of the group.

Ideally,  $P(\text{hit})$  should take into consideration the stochastic behavior of each TTL value, which is renewed every time a new copy of data is downloaded from the remote server. To achieve this, we would need to run stochastic simulations for caching, taking into consideration inputs, such as cacheable data volume, probability of data being cached, probability of data being accessed, and different TTL values to test for static and semi-static data. However, this is not the approach we are taking.

Our assumption is that the TTL is much greater than the inter-access time. This means that only the most-recently accessed items will be in the cache and the only reason to retrieve a data from the remote cloud is because the needed object is not in the cache. Therefore, we would divide the data type into two categories, cacheable and non-cacheable. The  $P(\text{hit})$  values from the Zipf distribution, shown in Figure 6, would be used in our simulation calculations. The Zipf distribution is a popularity model, where the probability of an object being requested is proportional to the rank of that object [4]. Figure 6 uses this Zipf popularity model, where hit ratios are plotted as a function of cache size for five kilobyte objects with four Alpha values (a higher Alpha value means that associated objects are much more popular). This graph does not take into account the expiration times of objects. It provides us reasonable hit ratios that we require to run our simulation.

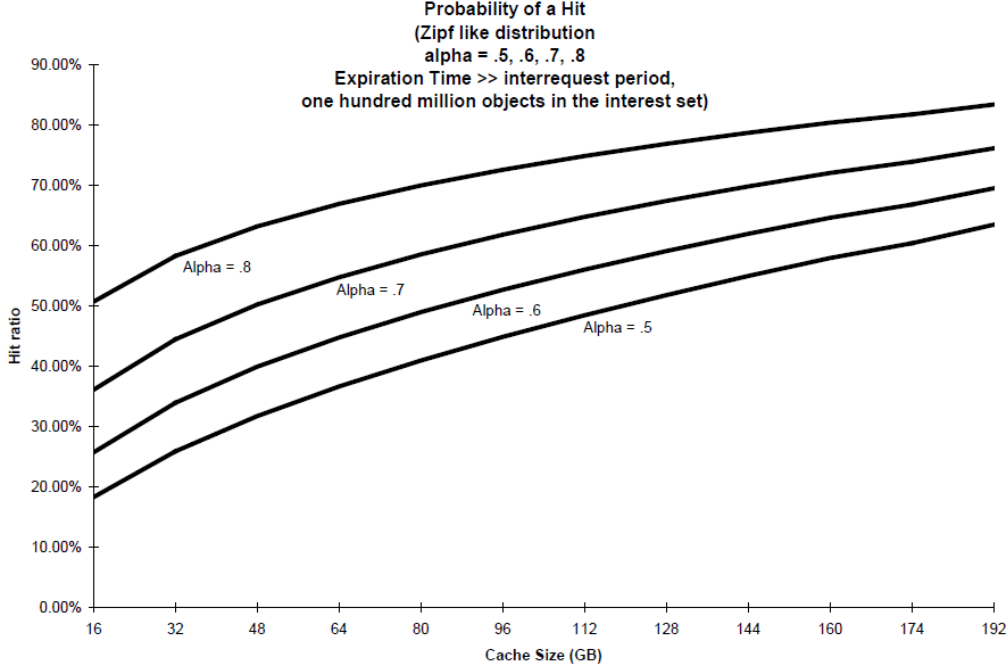


Figure 6. Hit ratio against cache size (from [4]).

The corresponding data volume of each category is determined simply by the following formulas:

$$\eta = \%Cacheable \times Total\_data\_volume \quad (2)$$

$$\mu = (1 - \%Cacheable) \times Total\_data\_volume \quad (3)$$

where  $\eta$  is the Cacheable data volume and  $\mu$  is the non-Cacheable data volume.

We define the total data requirement, TR, to be the data demand that will go through the SATCOM link, such that

$$TR = \mu + \eta \times (1 - P(\text{hit})) \quad (4)$$

where  $\eta$  and  $\mu$  are as defined above.

With values of Cacheability,  $P(\text{hit})$  and TR established, we can get the average response time,  $\zeta$ , using the following formula.

$$\zeta = \delta \times P(\text{hit}) + (\delta + \psi) \times P(\text{miss}) \quad (5)$$

where  $\delta$  is the average local access time,  $\psi$  is the average remote access time, and  $P(\text{miss})$  is the miss ratio defined as  $(1 - P(\text{hit}))$ . The average remote access time is defined as the time taken for the requesting node to receive a reply from the data origin after the request is made.

The value of  $\psi$  can be expressed as a function of TR and Bandwidth (BW):

$$\psi = f(TR, BW) \quad (6)$$

and can be obtained via QualNet [14] by entering the total data demand rate, TR, as input to the SATCOM model. With that, average response time,  $\zeta$ , becomes:

$$\zeta = \delta \times P(\text{hit}) + ((\delta + f(\text{TR}, \text{BW})) \times (1 - P(\text{hit}))) \quad (7)$$

The typical response time for accessing local cache,  $\delta$ , is between 30 to 35 milliseconds according to an article from ScaleOut [15]. Our experiment uses a fixed value of 30 milliseconds as  $\delta$ .

We adopted an incremental approach and prepared three test cases. The base case forms the baseline of the simulation; Case 1 includes the implementation of cache, while Case 2 includes the implementation of cloudlet.

#### A. Base Case without Cache

The base case models the scenario where caches are not implemented. The results from this base case form the baseline for our analysis. Since caches are not implemented, there is no cacheable data per se and the values for %cacheable and %non-cacheable are 0 percent and 100 percent, respectively. For the same reason, the hit ratio,  $P(\text{hit})$  is zero. Therefore, the TR for the base case will be the total data volume that is going to the remote cloud server via the SATCOM. With that, the generic formulas for TR and average response time,  $\zeta$ , presented in the previous section are reduced to:

$$\text{TR} = \text{Total\_data\_volume} \quad (8)$$

$$\zeta = \delta + f(\text{TR}, \text{BW}) \quad (9)$$

We use our QualNet model shown in Figure 7 to measure the time taken for the remote server to reply after the source node initiates the request.

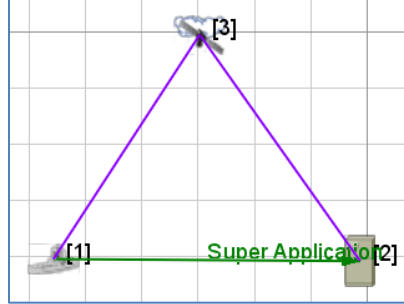


Figure 7. QualNet Model for base case without cache.

In our simulation, we vary the total data requirement, starting from 10 users to 60 users and increasing in steps of 10 users. We aim to find how the average response time will change with the increasing data requirement. Another parameter that we are varying is the available bandwidth of the SATCOM. By determining how the bandwidth affects the average response time will give us a good estimation of the minimum bandwidth requirement that we need for the given amount of data volume. This also gives us some idea of how the implementation of cache can overcome the effect on performance when operating under a limited bandwidth.

#### B. Case 1: Modeling with Local Cache

In this model, we assume that the naval platform has a local cache which would store some of the data objects. If the platform is requesting objects which are available in the local cache, the response time would be faster than requesting from a remote server or cloud. But since

we are not expecting everything in the Internet to be available in the local cache, the average response time of the requests would potentially be slower.

The QualNet model (from Figure 7) is modified to include a local cache at the node. The configuration of the node and the calculation of the response time are achieved using Excel. Using the result from the base case, we decide to keep bandwidth, BW, fixed at an optimal value of 2 Mbps, as the behavior with varying bandwidth is intuitive. Without considering the bandwidth, Equation (7) is potentially reduced to:

$$\zeta = \delta \times P(\text{hit}) + (\delta + f(\text{TR}) \times (1 - P(\text{hit}))) \quad (10)$$

The average response time is very much dependent on the TR and P(hit). Similarly, TR is directly proportional to the number of users using the network simultaneously, and we vary the TR from 10 to 60 users, in steps of 10 users, the same way as in the base case. In addition to that, we vary %cacheable and P(hit). For %cacheable, we vary from 30 percent to 70 percent in steps of 10 percent. As for P(hit), we vary from 10 percent to 80 percent in steps of 10 percent so that we can substantially cover the whole range of P(hit) for different Alpha values shown in Figure 6. Through the simulation runs, we aim to get some findings on how local cache affects the overall performance of the communications with respect to P(hit), cache size, popularity distribution, and number of users.

### C. Case 2: Modeling with Local Cache and Cloudlet

This mitigation takes into consideration two factors, the number of connections to the cloudlet node and the data requests by the nodes to the cloudlet node, then to the remote cloud. Building on Case 1, we evaluate whether implementing cloudlets will further improve the performance. All the test parameters remain the same, with the exception of the calculation of the average response time. A new formula is worked out with the following considerations.

When a source node makes a request, it will search its local cache for the data. If the node cannot find the data it seeks in its local cache, it will look for the data in the cloudlet node (Figure 8). When this occurs, it will be considered a miss on the source node, and the source's local access time and miss ratio as well as the inter-ship access time are taken into consideration. Similarly, the cloudlet node will search for the requested data in its local cache, and if it does not find the data, it will have to make a request to the destination cloud server. Now, the cloudlet's local access time and miss ratio are taken into consideration, and added to the source's initial response time. In both situations, the average local access time is the same as both nodes are treated independently. As the cloudlet node makes a request to the destination cloud server, the remote access time and the miss ratio needs to be taken into consideration in the formula. As a result, the formula becomes:

$$\zeta = \delta \times P(\text{hit}) + \{\beta + \delta + [\delta \times P(\text{hit}) + (\delta + \psi) \times P(\text{miss})]\} \times P(\text{miss}) \quad (11)$$

where  $\beta$  is the inter-ship access time.

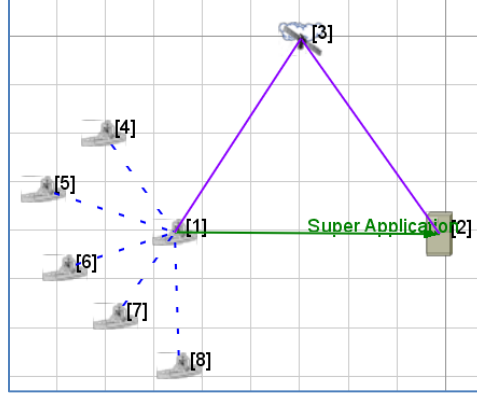


Figure 8. QualNet Model for base case with local cache and cloudlet.

Inter-ship access time is affected by a few parameters, namely the volume of data, the data rate, the LOS distance, and etc. Due to the classification of ship-to-ship communications, we were unable to ascertain these parameters. Instead, we made an estimation using propagation delay. Typically, the maximum LOS distance between the source node and the cloudlet node is 20km. Therefore, a two-way propagation delay is calculated to be approximately 130μsec.

$$\text{Propagation\_delay} = \frac{2 \times 20000}{3 \times 10^8} \approx 130 \mu\text{sec}$$

As compared to the local access time, this inter-ship access time is quite insignificant. So for Case 2, it is reasonable to assume that the inter-ship access time is negligible. We also assume that inter-ship communication is always available. As a result, the formula is reduced to:

$$\zeta = \delta \times P(\text{hit}) + \{\delta + [\delta \times P(\text{hit}) + (\delta + \psi) \times P(\text{miss})]\} \times P(\text{miss}) \quad (12)$$

The average remote access time is still dependent on TR. In this case, TR is obtained from the number of nodes connected to the cloudlet node. Let  $n$  be the number of connecting nodes. Effectively, by increasing the number of nodes, it is equivalent to increasing the number of users. For example, we assume that each node can support 10 users. When one node is connected to the cloudlet node ( $n = 1$ ) and  $P(\text{hit}) = 0$ , the TR is based on a maximum of 20 users, where 10 users belong to the cloudlet node and 10 users belong to the connecting node. When  $n = 2$  and  $P(\text{hit}) = 0$ , the TR is based on a maximum of 30 users, and so on and so forth. In general, the number of equivalent maximum number of users at the cloudlet node equals:

$$10 \times (n+1) \times (1-P(\text{hit})) \quad (13)$$

Through this experiment, apart from accessing the performance provided by the cloudlet implementation, we are also able to find the optimal/maximum number of nodes that can be connected to one cloudlet so that the available bandwidth can be optimized. The results are to be discussed in the next section.

## VI. Discussion of Results

Figure 9 shows the results of the base case, where the average response times in seconds, plotted against a range of SATCOM bandwidth from 1 to 2.5 Megabits per second (Mbps) for the base case without cache. Six curves, representing 10 to 60 users, are plotted in the same

graph as shown in the figure. The average response times are calculated using the average remote times collected from the QualNet. It is observed that there is a higher rate of improvement in the average response time when the bandwidth is increased from 1 Mbps to 1.5 Mbps. When there are more users (more load to the communication channel), the improvement seems to be more obvious. Based on the trend of the curve (left-hand side), we can infer that it is more significant to improve the bandwidth of the communications when network connectivity is limited. Intuitively, this matches with the expected behavior. When the bandwidth increases to 2 Mbps or higher, it is observed that the rate of improvement in the average response time becomes more gradual. For subsequent simulations, the bandwidth is fixed at 2 Mbps. This is reasonable because we can infer how bandwidth will affect the behavior of the performance by varying other parameters which are more interesting.

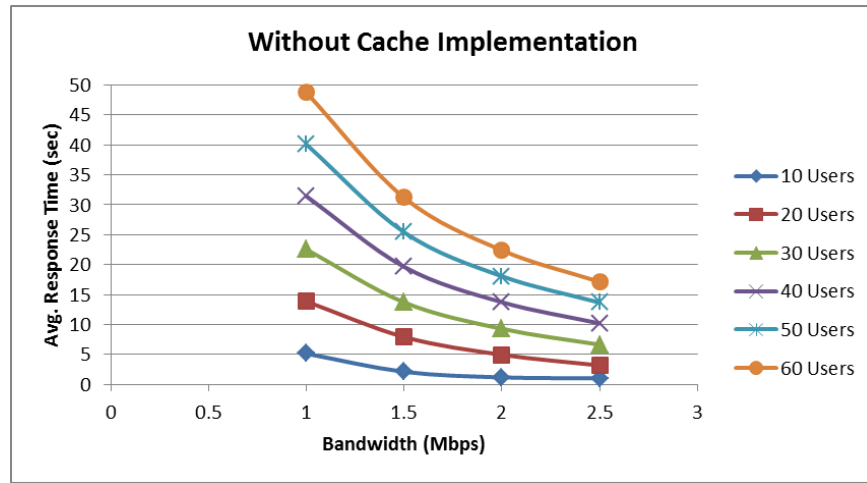


Figure 9. Average response time plots for base case (with no cache).

Figure 10 shows the effect of varying the percentage of cacheable data (hereafter, we refer to it as cache ratio) and number of users on the cloud response time for Case 1, where the naval ship has the ability to store data objects from the Internet so that the some of the data objects are available locally and there is no need to request it from the remote cloud. We varied the cache ratio and keep the other parameters,  $P(\text{hit})$  and number of users, constant. This way, we can observe the behavior specific to cache ratio. In Figure 10 (a), we see multiple curves plotted on the same graph for 10 users, each representing one  $P(\text{hit})$  value. Hence, we have eight curves for  $P(\text{hit})$ , ranging from 0.1 to 0.8. Figure 10 (b) shows a similar graph, but with the number of users fixed at 60 users. From the two graphs, we can observe that the cache ratio does not have a significant effect on average response time. For the case of 60 users, the cache ratio affects the response time slightly more but rather insignificantly. Hence, for the subsequent simulations, we would just look at 0.3 and 0.7 cache ratio (i.e., 30% and 70% cacheable data) so as to observe the results at the two extremes.

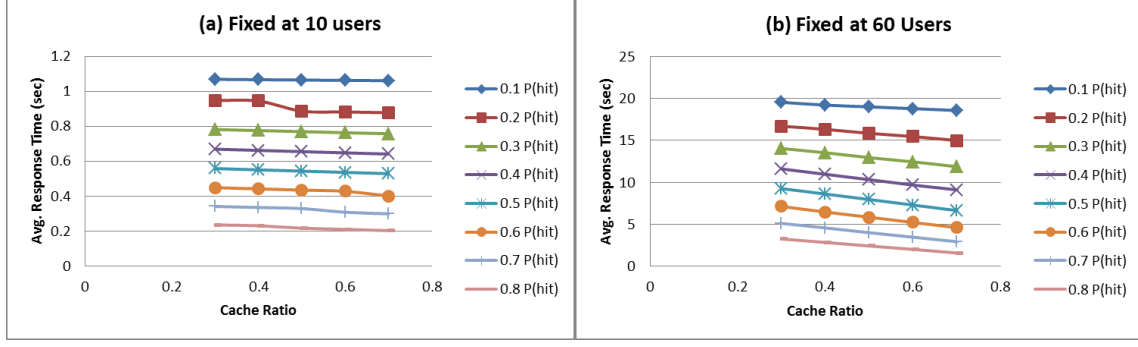


Figure 10. Average response time plots with (a) 10 users, (b) 60 users.

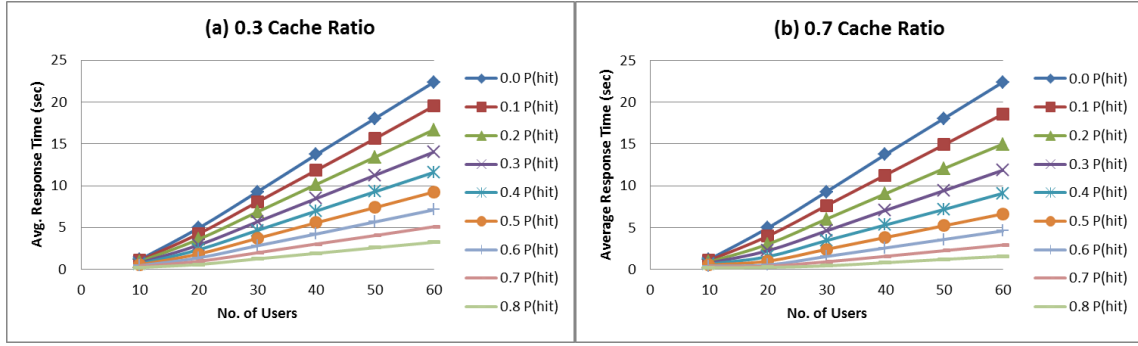


Figure 11. Varying the number of users with (a) 30%, (b) 70% cacheable data.

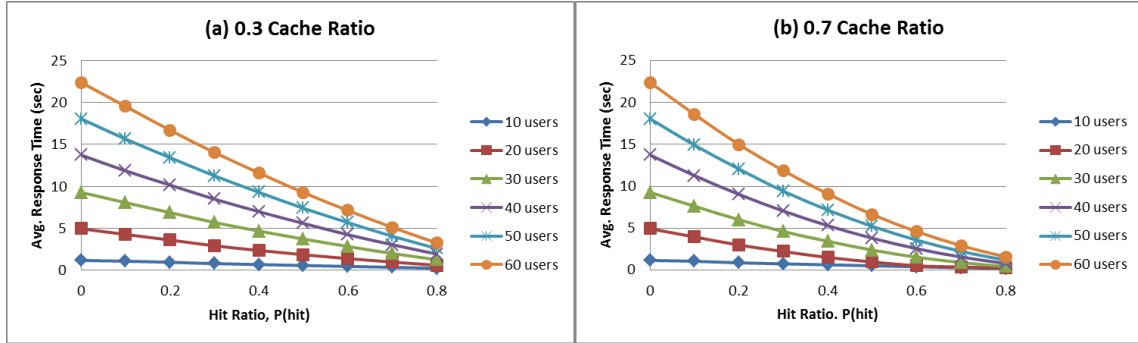


Figure 12. Varying  $P(\text{hit})$  with (a) 30%, (b) 70% cacheable data.

Figure 11 and 12 present two views of the experimental results with varying number of users and  $P(\text{hit})$ . Both figures show that varying the cache ratio between 0.3 and 0.7 does not have a significant impact on the average response time. Based on the above graphs, we can make the following observations:

- Increasing the number of user essentially increases the data load. By comparing the curves (taking note of  $P(\text{hit})=0.1$  and  $P(\text{hit})=0.8$ ), it is observed that the performance of the communications is greatly affected by the number of users when  $P(\text{hit})$  is 0.1 but not as much when  $P(\text{hit})$  is 0.8. From this we can conclude that compared with low  $P(\text{hit})$ , a high  $P(\text{hit})$  leads to less severe performance degradation as the volume of data (or the number of users) increases. Referring to Figure 6, we know that  $P(\text{hit})$  can be improved by increasing cache size or increasing the alpha value.



- Although the percentage of cacheable data (in the range of 30% to 70%) does not play a big part, we do observe that a higher cacheable ratio increases the rate of performance improvement with higher  $P(\text{hit})$ ; it is especially obvious in the case of 60 users. That is, the performance approached the asymptotic optimum more quickly when the hit ratio is higher, the result being due either to increased percentage of cacheable data or increased cache size.
- Based on the observation that higher  $P(\text{hit})$  results in greater performance improvement for higher data load (60 users versus 10 users), it is not recommended to improve the  $P(\text{hit})$  for relatively low data load conditions if the cost of doing that is high.
- While the simulations were done using bandwidth fixed at 2 Mbps, it is reasonable to infer that with higher bandwidth, the curves would just shift downwards, but the trend would remain the same. That is, as the offered load increases with respect to the capacity of the communications channel, the value of the cache, in terms of impact on average access time, increases exponentially.

Figures 13 and 14 show the effect of cloudlets on the performance of the data transmission. The number of users is fixed at 10 for the connecting node and cloudlet node in the simulations. Here, the variable  $n$  is the number of nodes that are connecting to the cloudlet node at the same time. Increasing the value of  $n$  is equivalent to increasing the data demand or the data load, since the equivalent number of users in a cloudlet of  $n$  node equals  $10 \times (n+1)$ . Similar to previous cases, the simulation is carried out with varying data loads and hit ratios. However, in this case, we assume that ship-to-ship communication is always available and the inter-ship access time is negligible. This is because we are only interested in the time required to fetch the data and not the ship's communication time. If inter-ship access time is to be taken into consideration, we foresee that this small time constant will cause the graphs of our results to exhibit a slight upward shift without changing the nature of the curves.

As before, further analysis is conducted on 0.3 and 0.7 cache ratios, which forms the lower and upper bound on the cache ratio, respectively. We compare the results for cases before and after cloudlets are implemented. The overall trend for the average response time is decreasing. This is desirable because the lower the response time, the better is the performance.

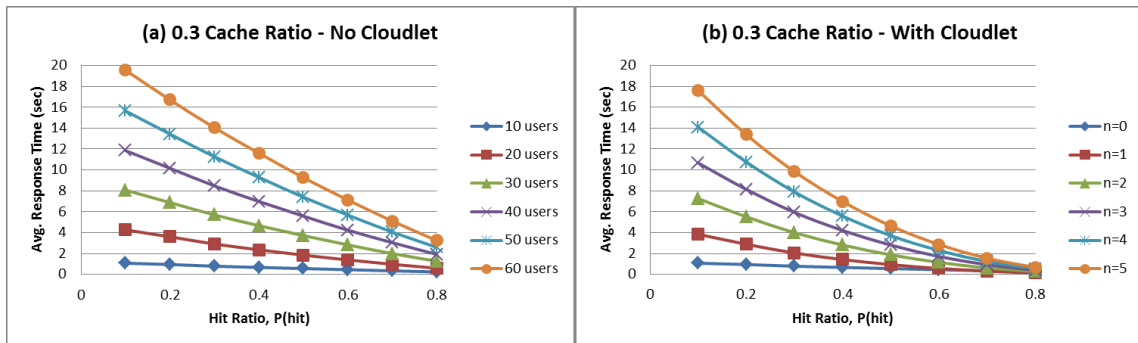


Figure 13. Average response time (with 0.3 Cache Ratio)  
 (a) *without* cloudlet (max. number of users =  $10 \times (n+1)$ ),  
 (b) *with* cloudlet (number of nodes =  $n$ ).

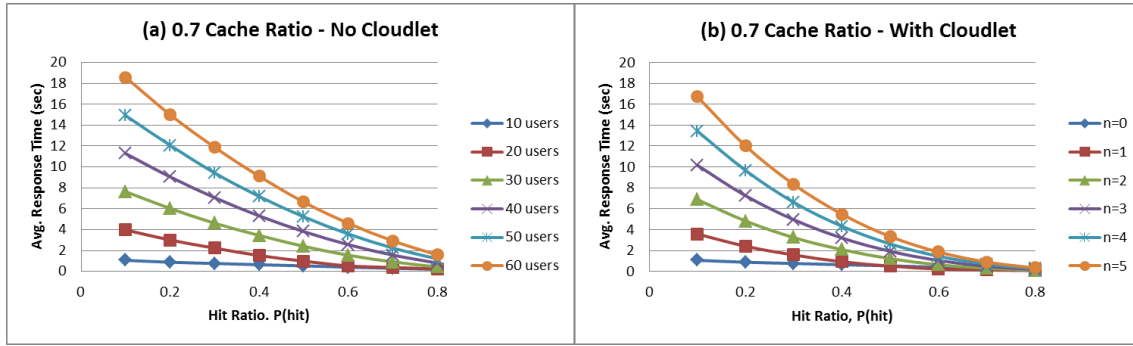


Figure 14. Average response time (with 0.7 Cache Ratio)  
 (a) *without* cloudlet (max. number of users =  $10 \times (n+1)$ ),  
 (b) *with* cloudlet (number of nodes =  $n$ ).

Having more nodes connecting to the cloudlet node will cause the data load to increase proportionally on the cloudlet node. Based on these graphs, we have the following observations:

- The slopes are steeper for the case with cloudlet as compared to the case without cloudlet.
- The improvement is more significant in the case of higher data requirements. The higher the number of nodes ( $n=5$ ) that are connected to the cloudlet node, the better the benefits.
- The response time tends to converge at the hit ratio of 0.8. Looking at the right end of the graphs (at  $P(\text{hit})=0.8$ ), the gaps between the response times for the different number of connecting nodes ( $n=1$  to 5) appear to be greater than when cloudlets are not implemented. Although this is not indicative that a 0.8 hit ratio is the optimal setting, the results further show that by implementing cloudlets we can improve the performance.

These observations highlight that the performance improvement is more significant when more nodes (indirectly more users and data loads) are leveraging the cloudlet. One possible explanation is that when the data objects are available in the cloudlet, more nodes ( $n=5$ ) connecting to the cloudlet would benefit from the time saved by accessing the cloudlet instead of the remote cloud. This suggests that the cloudlet solution will scale well for larger strike groups.

While this paper did not address the specific data exchange requirements within a strike group, it is reasonable to assume that a significant amount of data generated within the group is of interest to all members of the group. Thus, leveraging a cloudlet to cache that data or redirect it to appropriate members of the group underscores the point that the nodes are still able to have continuity in their operation even when each does not have direct satellite connectivity. However, the bottleneck caused by the cloudlet node is not studied in our research, so we cannot comment on the maximum number of nodes that can be connected to the cloudlet.

Looking from another perspective, the result is showing that the response time for a higher  $n$  can be as good as the response time of a lower  $n$  with higher  $P(\text{hit})$ . This is encouraging for the designer of the cloudlet to achieve higher  $P(\text{hit})$  especially for the case of high data load.

In the base case simulations, we can see that bandwidth is an important factor in the SATCOM. It is easy to increase the bandwidth in a simulation setup so as to improve the performance, but this is not always possible in a real-world situation. Bandwidth is usually fixed

or capped at a certain range and most of the time causes bottlenecks in the communications infrastructure. That is the main reason for keeping the bandwidth constant in our simulations so that we can focus our study on the cache and cloudlet. That said, any caching of data locally or at a cloudlet within the strike group reduces the demands on the satellite link, thereby improving the performance of the entire system.

## VII. Conclusions

Wildly fluctuating wireless bandwidth availability, intermittent connectivity, and unreliable connectedness (DIL connections) of SATCOM cause challenges for afloat platforms required to maintain connection with land-based clouds. Being able to exchange information with the cloud servers is very important to the support of U.S. Navy operations. To overcome some of these challenges, this paper proposed to supplement the cloud architecture with two strategies, local caches and strike group hosted cloudlets. Our study showed that the implementation of caching can indeed improve the response time of requests made by the users. We were able to show that the use of a cloudlet is able to further improve performance. The cloudlet can act as an alternative to the remote cloud when the direct connection to the satellite is down or the capacity of the link is limited with respect to the traffic load. This increases the availability of the communication network so that the operations can still move forward, although it might be in a degraded mode as compared to the direct connection via SATCOM.

While the results obtained were positive, additional work is needed to further verify the effectiveness of the strategies in real environments. Practical evaluations in the U.S Navy context are necessary, before these strategies can be put to actual use. This includes the usage of actual C2 data, as well as the integration of the inter-ship access time, particularly the expected transmission delays given actual traffic loads and system capacities. This information was unavailable to us due to its sensitivity.

Although we have made a reasonable assumption about the inter-ship delay being negligible, it is more complete to capture the delay in the formula for future work that follows. The inter-ship delay can be modeled dynamically with a moving naval ship. The data rate can also be modeled with the consideration of whether there is a collision medium or not.

While our work examined the case where there is only one cloudlet node, the scope can be extended to study whether all nodes can take on the role of the cloudlet node. This would be analogous to establishing an ad-hoc meshed topology. Such an ad-hoc mesh may allow for parallelism in cache searches. Searching in the local caches of all the cloudlet nodes first may further reduce the need to send requests to the remote cloud server, limiting dependence on a connection back to the remote cloud server via SATCOM and enhancing continuity when operating in a DIL environment. In addition, optimization can be conducted to find out a few things, for example, the maximum number of users per node, the optimal number of nodes per cloudlet node, and also the maximum number of cloudlet nodes that can be supported by a given amount of bandwidth. This will facilitate decision making in U.S. Navy operations, taking into account the tradeoffs between performance and load. We foresee that this could be achieved by tweaking the inputs and reworking the formulas used in the model that we have developed.

## References:

- [1] Alhamad, M., Dillon, T., Wu, C. & Chang, E. (2010). *Response Time for Cloud Computing Providers*. Paper presented at WAS2010, Paris, France, November 8–10, 2013. doi:10.1145/1967486.1967579.
- [2] Amazon Web Services (2012). *The Amazon CloudFront Developer Guide*, Amazon Web Services, Inc., 02 March 2012. Accessed on 2/3/2015: <http://docs.aws.amazon.com/AmazonCloudFront/2010-11-01/DeveloperGuide/Introduction.html>
- [3] Agarwal, A. (2008). *How to Setup Amazon S3 with CloudFront as a Content Delivery Network*, Digital Inspiration, 18 November 2008. Accessed on 2/3/2015: <http://www.labnol.org/internet/setup-content-delivery-network-with-amazon-s3-cloudfront/5446/>
- [4] Beaumont, L. R. (2000, March). *Calculating Web Cache Hit Ratios*. Accessed on 2/3/2015: <http://www.content-networking.com/papers/web-caching-zipf.pdf>.
- [5] Chen, H. (2013, March 20). The AWS Olympics: Speed Testing Amazon EC2 and S3 Across Regions. Accessed on June 4, 2014. Retrieved from <http://www.takipiblog.com/2013/03/20/aws-olympics-speed-testing-amazon-ec2-s3-across-regions/>
- [6] Dey, S., Liu, Y., Wang, S. & Lu, Y. (2013). Addressing Response Time of Cloud-based Mobile Applications. Paper presented at MobileCloud'13, Bangalore, India, July 29–August 1, 2013. doi:10.1145/2492348.2492359
- [7] Hennessy, J. L., & Patterson, D. A. (2012). *Computer architecture: A Quantitative Approach*. Waltham, MA: Elsevier.
- [8] Lopic, S., Ching, D., Labbe, I., Jordan, M., Meagher, C., Chong, L., Dumoulin, S. & Thompson, R. (2013). *Maritime Operations in Disconnected, Intermittent, and Low-bandwidth Environments*. San Diego, CA: Space and Naval Warfare Systems Center Pacific.
- [9] Osa, J., Castello, R., Radulovich, J., Gillcrist, B., & Finocchiaro, C. (2004, April). Distributed Positioning, Navigation and Timing (DPNT). In *Proceedings of IEEE 2004 Position Location and Navigation Symposium*. Piscataway, NJ: IEEE.
- [10] Perkins, R., Dejesus, F., Durham, J., Hastings, R., & McDonnell, J. (2013, June). *C2 Data Synchronization in Disconnected, Intermittent, and Low-bandwidth (DIL) Environments*. Paper presented at 18th ICCRTS, June 19–21, 2013, Alexandria, VA.
- [11] Satyanarayanan, M., Bahl, P., Caceres, R., & Davies, N. (2009). The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing*, 8(4), 14–23.
- [12] Satyanarayanan, M., Lewis, G., Morris, E., Simanta, S., Boleng, J., & Ha, K. (2013). The Role of Cloudlets in Hostile Environments. *IEEE Pervasive Computing*, 12(4), 40–49.
- [13] Sandvine (2014). *Global Internet Phenomena Report 1H2014*. Retrieved from <https://sandvine.com/downloads/general/global-Internet-phenomena/2014/1h-2014-global-Internet-phenomena-report.pdf>

- [14] Scalable Network Technologies, Inc. (2013). *QualNet 7.1 Programmer's Guide*. Culver City, CA: Scalable Network Technologies.
- [15] ScaleOut. (2007). *Distributed Caching: Fast Response Time and Scalability*. Retrieved from [http://h21007.www2.hp.com/portal/download/product/24028/SOSS\\_Performance\\_06-07\\_1207761260469.pdf](http://h21007.www2.hp.com/portal/download/product/24028/SOSS_Performance_06-07_1207761260469.pdf).
- [16] Wee, T.J., & Ling, Y.X. (2014). *Mobility And Cloud: Operating In Intermittent, Austere Network Conditions*, Master's Thesis, Naval Postgraduate School, Monterey, CA.
- [17] Wessels, D. (2001). *Web Caching*. Sebastopol, CA: O'Reilly Media.